

Analisis Kinerja Deteksi Algoritma FAST dan Algoritma MSER pada Citra Digital Berbasis Marker

Analysis of FAST Algorithm and MSER Algorithm Detection Performance on Marker-Based Digital Image

Dwi Anugrahita^{1*}, Rini Mayasari², Susilawati³

^{1,2,3}Progam Studi Teknik Informatika, Fakultas Ilmu Komputer

Universitas Singaperbangsa Karawang

Jl. HS. Ronggo Waluyo, Puseurjaya, Karawang, 41363, Indonesia

email: ¹dwi.16076@student.unsika.ac.id, ²rini.mayasari@staff.unsika.ac.id,

³susilawati.sobur@staff.unsika.ac.id

ABSTRAK

DOI;
10.30595/jrst.v5i2.7796

Histori Artikel:

Diajukan:
08/07/2020

Diterima:
11/08/2022

Diterbitkan:
26/08/2022

Augmented Reality (AR) membutuhkan algoritma yang baik dan tahan terhadap gangguan-gangguan yang dapat terjadi saat proses deteksi marker seperti perubahan pencahayaan, perubahan rotasi marker, dan blur pada kamera. Algoritma *Feature from Accelerated Segment Test* (FAST) dan algoritma *Maximally Stable Extremal Regions* (MSER) merupakan algoritma yang kerap dipakai sebagai metode pendeteksi *marker* pada AR. Tujuan penelitian ini yaitu untuk menganalisis kinerja algoritma FAST dan algoritma MSER terhadap kemampuan dan kecepatannya untuk mendeteksi dan mengekstraksi fitur-fitur pada citra serta ketahanannya terhadap gangguan pada citra yang digunakan untuk *marker* pada AR, dan menunjukkan algoritma yang lebih baik untuk aplikasi AR antara algoritma FAST dan algoritma MSER. Penelitian ini menggunakan dua set citra; *2D Template Marker* dan *Image Marker* dengan 10 gambar untuk masing-masing set dan modifikasi pada citra seperti; perubahan intensitas cahaya, rotasi, dan *blur* dengan parameter: (1) jumlah fitur yang terdeteksi, (2) waktu deteksi dan ekstraksi fitur, (3) persentase banyaknya fitur yang berhasil dicocokkan, dan (4) waktu pencocokan fitur. Berdasarkan keempat parameter, algoritma FAST memiliki pemrosesan yang lebih cepat terhadap deteksi *marker*, sedangkan algoritma MSER memiliki proses pendeteksian *marker* yang lebih stabil terhadap perubahan yang terjadi baik pada kamera atau *marker*.

Kata Kunci: *Augmented Reality, Feature from Accelerated Segment Test, Maximally Stable Extremal Regions, Penanda AR*

ABSTRACT

Augmented Reality (AR) requires a good algorithm which good to disturbance that may occur during the marker detection process such as changes in lighting, blur on the camera, and rmarker rotation changes. Feature From Accelerated Segment Test (FAST) algorithm and Maximally Stable Extremal Regions (MSER) algorithm is an image detection algorithm that often used as a marker detection method in AR. Main purpose of this study is to analyze the performance of the FAST algorithm and MSER algorithm, its ability, and speed to detect and extract features image, its robustness, and show a better algorithm for AR applications among the FAST algorithm and MSER algorithm. This study use two image set; *2D Template Marker* and *Image Marker*, with 10 images each sets and changes to the image such as; changes in light intensity, rotation, and

blur with parameters: (1) number of detected and extracted features, (2) time elapsed during features detection and extraction, (3) percentage of feature points successfully matched, and (4) time elapsed during feature matching.

Keywords: Augmented Reality, Feature from Accelerated Segment Test, Maximally Stable Extremal Regions, AR Marker

1. PENDAHULUAN

Proses deteksi *marker* pada *Augmented Reality* (AR) sangat penting untuk AR yang menggunakan *marker*, semakin cepat *marker* terdeteksi maka semakin cepat pula objek virtual ditampilkan. Dengan objek virtual yang muncul sesaat setelah *marker* terdeteksi, maka tercapai salah satu karakteristik dari AR, yakni *real time*. Proses deteksi *marker* dapat dipengaruhi oleh spesifikasi perangkat, pencahayaan, serta jarak kamera dengan *marker*. Selain itu, proses deteksi *marker* bergantung pada algoritma yang digunakan untuk mendeteksi fitur-fitur pada citra berbasis *marker* yang digunakan pada AR.

Algoritma deteksi fitur sudah banyak dikembangkan oleh peneliti-peneliti terdahulu. Rosten dan Drummond (Rosten & Drummond, 2006) pada tahun 2006 memperkenalkan algoritma *Feature from Accelerated Segment Test* (FAST) yakni algoritma deteksi fitur berupa sudut pada citra. J. Matas, O. Chum, M. Urban, dan T. Pade (2002) memperkenalkan algoritma *Maximally Stable Extremal Regions* (MSER) sebagai algoritma deteksi fitur *blob* pada citra.

Penggunaan algoritma MSER dan penyederhanaan kontur memberikan akurasi yang lebih tinggi, lebih kuat untuk *marker* yang berupa tulisan tangan dengan perubahan pose kamera, dan membutuhkan komputasi lebih sedikit daripada metode lainnya (Thanabornvornwiwat & Patanukhom, 2018). Algoritma FAST, dan MSER mendeteksi lebih sedikit fitur pada citra daripada algoritma Harris, namun algoritma FAST memiliki kecepatan yang tinggi dibandingkan dengan algoritma lainnya (Bhushan Verma & Chandran, 2016). Selanjutnya studi perbandingan algoritma SIFT, MSER, FAST dan SURF untuk mengenali gerakan tangan menunjukkan bahwa algoritma MSER, FAST, dan SURF bekerja lebih dengan waktu yang lebih efisien dibandingkan dengan algoritma SIFT (Zafar, Berns, & Rodić, 2017). Proses deteksi *marker* dengan metode algoritma FAST dapat dipengaruhi oleh spesifikasi perangkat, pencahayaan, ukuran *marker*, dan ketebalan garis yang digunakan untuk *marker* (Wahyudi, Harianto, & Setyati, 2019). Penggunaan algoritma

MSER untuk mendeteksi teks pada kartu identitas menunjukkan bahwa algoritma MSER berkinerja dengan cukup baik dalam akurasi, presisi, maupun *recall* (Purba, Harjoko, & Wibowo, 2019).

Algoritma FAST dan algoritma MSER adalah algoritma yang digunakan dalam pengoperasian citra untuk mendeteksi ciri-ciri tertentu pada citra yang selanjutnya digunakan untuk *image processing*. Algoritma deteksi fitur citra yang baik adalah algoritma yang memiliki karakteristik *robustness*, *repeatability*, *accuracy*, *generality*, *efficiency*, dan *quantity*. Proses deteksi *marker* pada AR dipengaruhi oleh gangguan-gangguan terhadap citra seperti perubahan rotasi, *blur*, perubahan intensitas cahaya atau kecerahan citra, dan lain-lain. Proses deteksi *marker* yang lama berpengaruh pada aplikasi AR yang membuat objek virtual tidak muncul secara langsung, artinya tidak *real time*. Untuk dapat membuat sebuah aplikasi AR yang *real-time*, dibutuhkan algoritma yang tepat, yaitu algoritma yang cepat serta tahan terhadap perubahan yang dapat terjadi saat proses deteksi *marker* pada kamera, seperti perubahan pencahayaan, rotasi, dan *blur*. Berdasarkan penelitian yang telah dilakukan sebelumnya, algoritma FAST memiliki kecepatan yang tinggi, sedangkan algoritma MSER memberikan akurasi yang lebih baik dibandingkan algoritma deteksi fitur lainnya. Maka untuk mengetahui diantara algoritma FAST dan algoritma MSER untuk aplikasi AR, diperlukan pengujian terhadap kinerja deteksi kedua algoritma untuk *marker* AR.

Berdasarkan latar belakang masalah tersebut, penelitian ini dilakukan untuk menganalisis terhadap kinerja deteksi algoritma FAST dan algoritma MSER terhadap kemampuan dan kecepatannya untuk mendeteksi dan mengekstraksi fitur-fitur pada citra digital berbasis *marker* yang digunakan untuk AR, dan ketahanannya terhadap gangguan pada citra digital seperti perubahan pencahayaan, rotasi, dan *blur* untuk menunjukkan algoritma mana yang lebih baik untuk aplikasi AR. Sehingga dari hasil penelitian ini dapat diketahui dengan jelas kinerja dari algoritma FAST dan algoritma MSER

pada tahapan pendeteksian fitur dalam teknologi AR.

2. METODE PENELITIAN

Penelitian ini menggunakan metode penelitian eksperimental yang bertujuan untuk menguji pengaruh suatu variabel tetap terhadap variabel lain. Metode penelitian eksperimental dipilih karena penelitian yang dilakukan bertujuan untuk menganalisis kinerja deteksi dari algoritma FAST dan algoritma MSER terhadap citra asli dan citra manipulasi (perubahan pencahayaan, rotasi, dan *blur*). Adapun manipulasi terhadap gambar yang akan diuji adalah sebagai berikut.

Tabel 1. Manipulasi Terhadap Citra

Kecerahan	Rotasi	Blur
+100	90°	1px
+50	180°	
-50		
-100		

Tabel 1. menunjukkan manipulasi terhadap gambar atau citra yang akan diuji. Masing-masing citra diberikan manipulasi kecerahan +100; +50; -50; dan -100, kemudian diberikan Rotasi sebesar 90° dan 180°, lalu diberikan *blur* sebesar 1px.

2.2 Alat Bahan

Untuk mendukung penelitian ini, diperlukan perangkat keras dan perangkat lunak sebagai berikut.

1. Prosesor AMD Ryzen 5 2500U dengan AMD Radeon Vega Mobile Gfx 2.00 GHz,
2. RAM 8GB,
3. Harddisk 1 TB,
4. Sistem Operasi Windows 10 Home Single Language,
5. MATLAB R2016a dengan *tools image processing* untuk pengkodean,
6. Photoscape dan Photoshop CC 2019 untuk manipulasi gambar, dan
7. Vuforia SDK untuk mendapatkan *rating marker*.

Citra berbasis *marker* yang digunakan berjenis 2D *Template Marker* dan *Image Marker*. Citra berbasis *marker* yang digunakan untuk pengujian diambil berdasarkan *purposive sampling*, yang merupakan metode penarikan sampel berdasarkan kriteria-kriteria tertentu. Adapun citra berbasis *marker* digunakan apabila memenuhi kriteria sebagai berikut.

1. Gambar berdimensi 400 x 400 piksel dengan resolusi 72 dpi.
2. Gambar memiliki *rating* Vuforia minimal 3 bintang.

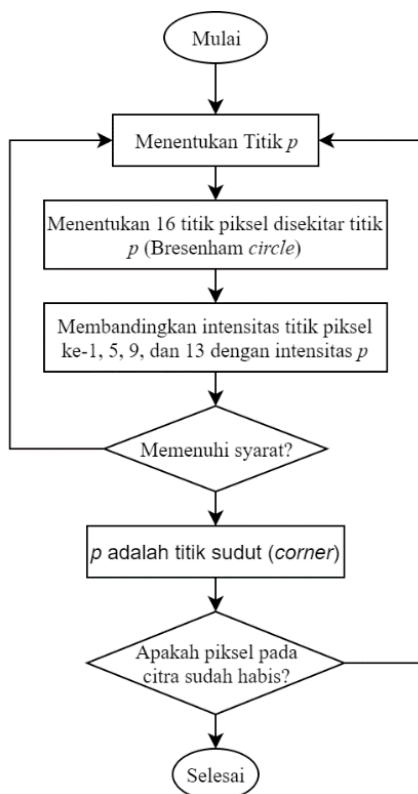
Tabel 2. Kriteria Gambar

No.	Kriteria	Sampel
1.	Gambar berdimensi 400 x 400 piksel dengan resolusi 72 dpi	20
2.	Gambar memiliki <i>rating</i> Vuforia minimal 3 bintang	20
Jumlah sampel penelitian		20

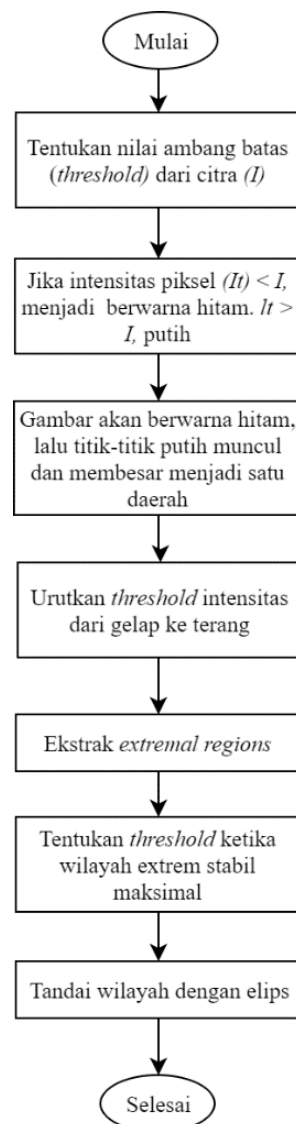
Setelah sampel gambar yang akan diuji didapatkan dan memenuhi kriteria berdasarkan **Tabel 2**, dilakukan 3 perlakuan berbeda untuk masing-masing gambar berdasarkan **Tabel 1**.

2.2. Pengkodean

Pengkodean atau pembuatan program pengujian untuk penelitian ini menggunakan MATLAB dengan mengimplementasikan algoritma FAST dan algoritma MSER. Citra yang akan diuji terdiri dari citra asli dan citra manipulasi. Fitur-fitur yang dimiliki oleh kedua citra kemudian akan dideteksi dan diekstraksi menggunakan algoritma FAST dan algoritma MSER, lalu fitur-fitur yang telah diekstraksi pada kedua citra akan dicocokkan satu sama lain menggunakan fungsi yang sudah ada pada MATLAB.

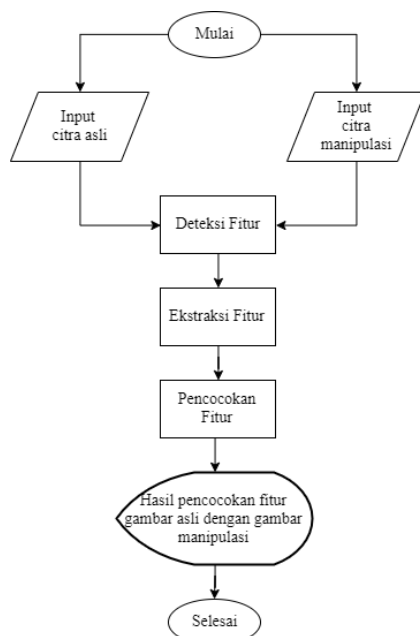


Gambar 1. Diagram Alir Algoritma FAST



Gambar 2. Diagram Alir Algoritma MSER

Tahapan utama dari sistem yang dibuat terdiri dari; (1) *input* citra, (2) deteksi fitur, (3) ekstraksi fitur, dan (4) pencocokan fitur. Citra yang *diinput* berupa citra asli dan citra manipulasi, kemudian akan dideteksi dan diekstraksi fitur-fitur pada citra menggunakan algoritma FAST dan algoritma MSER, dan fitur-fitur yang sudah diekstraksi pada kedua citra akan dicocokkan satu sama lain menggunakan fungsi yang sudah ada pada MATLAB. Diagram alir dari program pengujian adalah sebagai berikut.



Gambar 3. Diagram Alir Program Pengujian

2.3. Uji Kelayakan Marker

Rating bintang dari *marker* berkisar antara 1 sampai dengan 5 bintang. *Marker* diuji kelayakannya menggunakan Vuforia Engine. Walaupun *marker* dengan *rating* rendah (1 atau 2 bintang) biasanya dapat dideteksi dan dilacak dengan baik, gunakan *marker* dengan *rating* lebih dari 2 bintang untuk mendapatkan hasil pelacakan yang stabil. Target *marker* dideteksi berdasarkan fitur-fitur yang diekstraksi dari gambar kemudian dibandingkan secara langsung dengan *marker* yang tertangkap oleh kamera. Dipilih 10 gambar hitam-putih dan 10 gambar berwarna untuk diuji kelayakannya menjadi *marker* AR. Untuk mendapatkan kinerja deteksi dan pelacakan terbaik dari Vuforia Engine, target *marker* harus memiliki atribut sebagai berikut. Citra berbasis *marker* yang digunakan dirancang sedemikian rupa agar memiliki atribut-atribut sebagai *marker* yang ideal untuk mendapatkan *rating* Vuforia yang tinggi.

Tabel 3. Marker Ideal

Atribut	Contoh
Kaya akan detail	Adekan jalanan, sekelompok orang, kolase dan campuran item, dan adegan olahraga.
Memiliki kontras yang bagus	Daerah terang dan gelap, dan cukup terang.

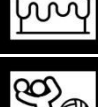
Tidak ada pola berulang

Format

Lapangan berumput, sekumpulan rumah dengan jendela yang identik, dan *checkerboard*.
PNG dan JPG 8 atau 24-bit; kurang dari 2 MB; JPG harus RGB atau *grayscale* (tanpa CMYK).

Hasil *rating* dan ID *marker* diperoleh dengan mengirimkan gambar *marker* ke situs developer Vuforia, yang ditunjukkan pada tabel-tabel sebagai berikut

Tabel 4. Uji Kelayakan pada Template Marker

Nama	Gambar	Tip	Rating
Temp1		Single Image	5
Temp2		Single Image	5
Temp3		Single Image	5
Temp4		Single Image	5
Temp5		Single Image	4
Temp6		Single Image	5
Temp7		Single Image	5
Temp8		Single Image	5
Temp9		Single Image	5
Temp10		Single Image	4

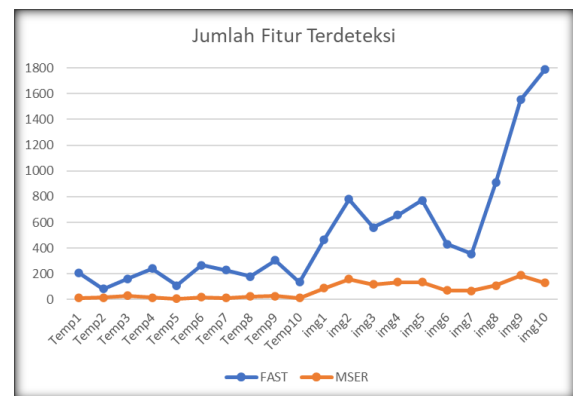
Tabel 5. Uji Kelayakan pada *Template Marker*

Nama	Gambar	Tippe	Rating
img1		Single Image	5
img2		Single Image	3
img3		Single Image	5
img4		Single Image	5
img5		Single Image	5
img6		Single Image	5
img7		Single Image	5
img8		Single Image	4
img9		Single Image	5
img10		Single Image	3

3. HASIL DAN PEMBAHASAN

3.1. Jumlah Fitur Terdeteksi

Jumlah fitur terdeteksi oleh algoritma FAST dan algoritma MSER pada citra asli ditunjukkan pada diagram garis sebagai berikut.

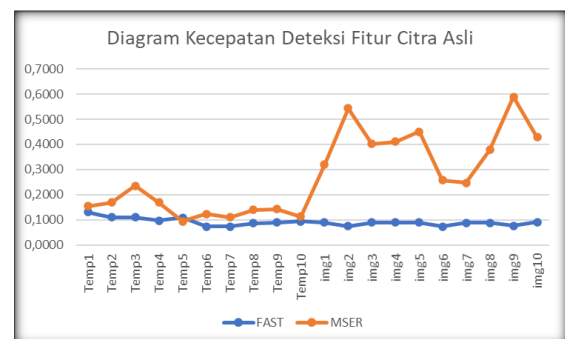


Gambar 4. Diagram Garis Jumlah Fitur Terdeteksi pada Citra Asli

Diagram Garis pada **Gambar 4** menunjukkan bahwa algoritma FAST mendeteksi lebih banyak fitur dengan rata-rata 192 fitur pada set 2D *Template Marker* terdeteksi dan rata-rata 827 fitur pada set *Image Marker* terdeteksi. Sedangkan algoritma MSER rata-rata mendeteksi sebanyak 18 fitur untuk set 2D *Template Marker* dan 120 fitur pada set *Image Marker*.

3.2. Kecepatan Deteksi dan Ekstraksi Fitur

Kecepatan deteksi dan ekstraksi fitur dari algoritma FAST dan algoritma MSER terhadap citra asli dapat dilihat pada diagram garis sebagai berikut.

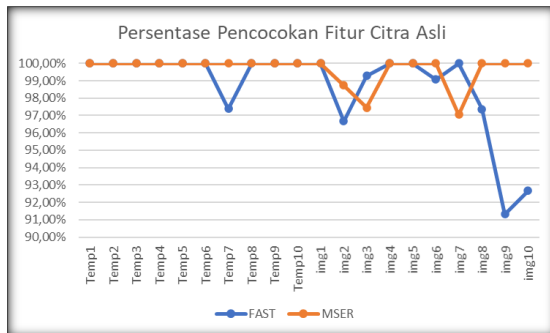


Gambar 5. Diagram Garis Kecepatan Deteksi dan Ekstraksi Fitur Citra Asli

Berdasarkan Diagram Garis pada **Gambar 5**, algoritma FAST memiliki kecepatan deteksi dan ekstraksi lebih cepat dengan *range* 0,07 – 0,1 detik dan rata-rata kecepatan deteksi dan ekstraksi fitur 0,09 detik. Sedangkan algoritma MSER membutuhkan waktu antara 0,09 – 0,5 detik dengan rata-rata kecepatan deteksi dan ekstraksi fitur 0,27 detik.

3.3. Persentase Pencocokan Fitur

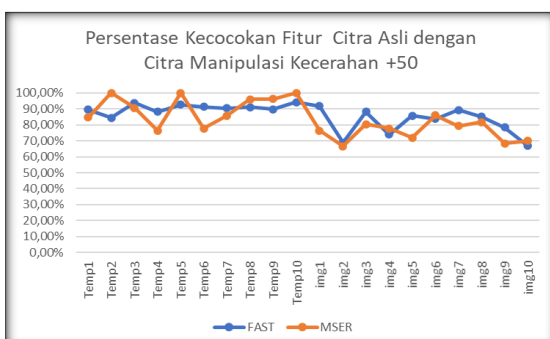
Dalam penelitian ini dilakukan beberapa pengujian pencocokan fitur terhadap gambar yang mendapatkan perlakuan berbeda-beda, yaitu pencocokan fitur citra asli, pencocokan fitur citra asli dengan citra perubahan kecerahan, pencocokan fitur citra asli dengan citra perubahan rotasi, dan pencocokan fitur citra asli dengan citra *blur*. Berikut hasil pencocokan banyaknya Fitur Citra Asli menggunakan algoritma FAST dan algoritma MSER.



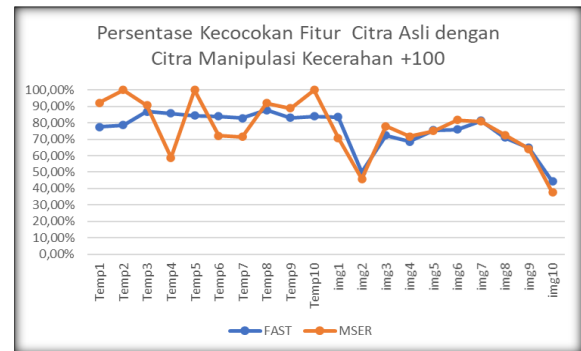
Gambar 6. Diagram Garis Persentase Pencocokan Fitur Citra Asli

Diagram Garis Persentase Pencocokan Fitur Citra asli pada **Gambar 6** menunjukkan bahwa algoritma MSER memiliki persentase pencocokan fitur lebih baik daripada algoritma FAST dengan *range* antara 97% – 100%. Sedangkan algoritma FAST memiliki persentase pencocokan fitur dengan *range* antara 91,3% – 100%.

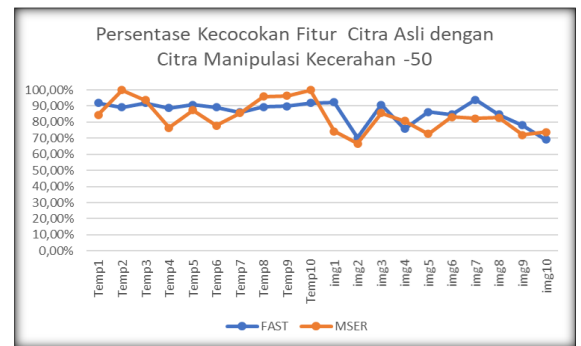
Kemudian dilakukan pencocokan fitur citra asli dengan citra hasil perubahan kecerahan. Perubahan kecerahan gambar yaitu sebesar +50, +100, -50, dan -100 untuk mengetahui seberapa baik kinerja kedua algoritma terhadap perubahan pencahayaan yang terjadi. Berikut hasil pencocokan fitur citra asli dengan citra hasil perubahan kecerahan.



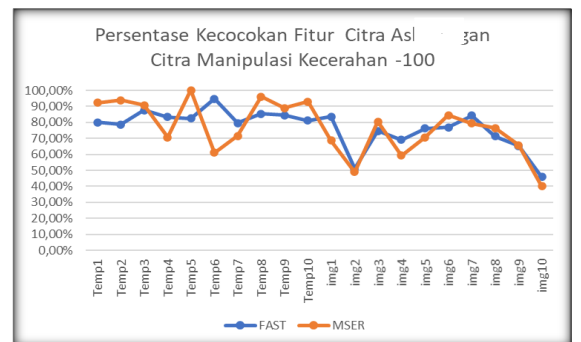
(a)



(b)



(c)

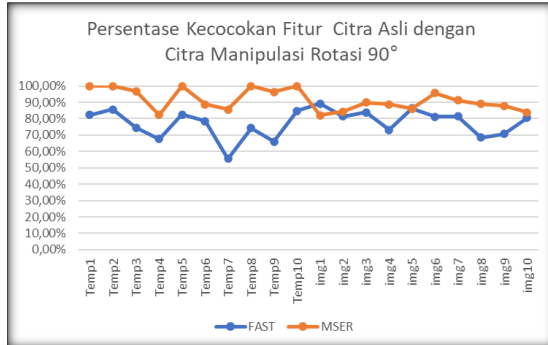


(d)

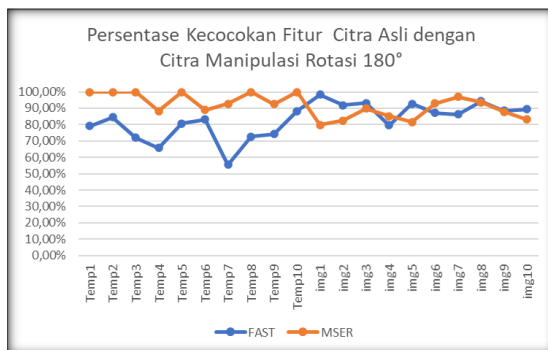
Gambar 7. Diagram Garis Persentase Pencocokan Fitur Citra Asli dengan Citra Manipulasi Kecerahan: (a) +50, (b) +100, (c) -50, (d) -100

Pada perubahan kecerahan +100 dan -100, persentase kecocokan fitur kedua algoritma cenderung tidak stabil dan menurun. Sedangkan pada perubahan kecerahan +50 dan -50, persentase kecocokan fitur kedua algoritma terlihat lebih stabil. Kedua algoritma dapat bekerja dengan baik terhadap perubahan pencahayaan.

Selanjutnya adalah pencocokan fitur citra asli dengan citra hasil perubahan rotasi. Perubahan rotasi gambar yaitu sebesar 90° dan 180° . Berikut hasil pencocokan fitur citra asli dengan citra hasil perubahan rotasi.



(a)

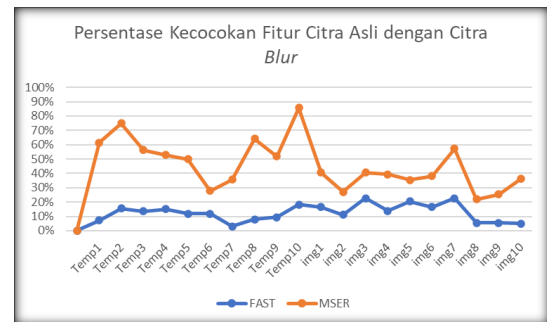


(b)

Gambar 8. Diagram Garis Persentase Kecocokan Fitur Citra Asli dengan Citra Manipulasi Rotasi: (a) 90° , (b) 180°

Berdasarkan Diagram Garis pada **Gambar 8**, perubahan rotasi pada gambar mempengaruhi persentase kecocokan fitur pada gambar. Perubahan rotasi 180° kedua algoritma memiliki persentase kecocokan yang fitur lebih baik daripada persentase kecocokan fitur pada perubahan rotasi 90° .

Terakhir adalah pencocokan fitur citra asli dengan citra yang sudah diberikan *blur* sebesar 1px untuk mengetahui seberapa baik kinerja kedua algoritma terhadap perubahan fokus yang terjadi. Berikut hasil pencocokan fitur citra asli dengan citra *blur*.

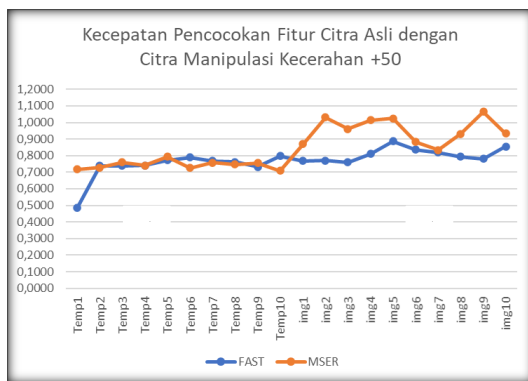


Gambar 9. Diagram Garis Persentase Kecocokan Fitur Citra Asli dengan Citra Blur

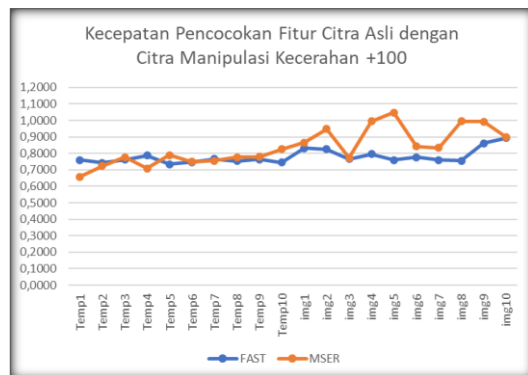
Berdasarkan Diagram Garis pada **Gambar 9**, kedua algoritma tidak bekerja cukup baik terhadap citra yang *blur*. Namun berdasarkan diagram garis diatas, kinerja algoritma MSER lebih baik daripada algoritma FAST karena rata-rata pencocokan fitur hampir mendekati 50%.

3.4. Persentase Pencocokan Fitur

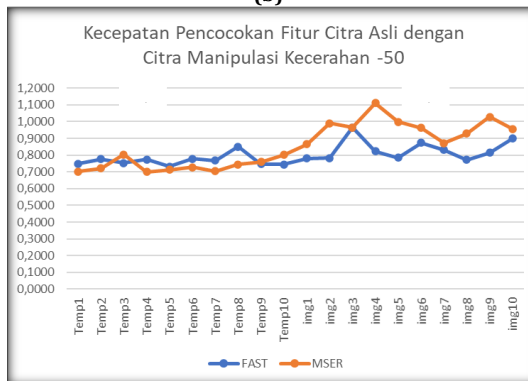
Pengujian tingkat kecepatan pencocokan fitur pada algoritma FAST dan algoritma MSER dilakukan terhadap citra asli dengan citra yang telah dilakukan perubahan terhadap kecerahan, rotasi, dan *blur*. Hal ini dilakukan untuk melihat seberapa cepat kinerja algoritma FAST dan algoritma MSER dalam melakukan pencocokan fitur terhadap citra asli dan citra manipulasi. Kecepatan pencocokan fitur dari algoritma FAST dan algoritma MSER terhadap citra asli dan citra manipulasi dapat dilihat pada diagram garis dibawah ini.



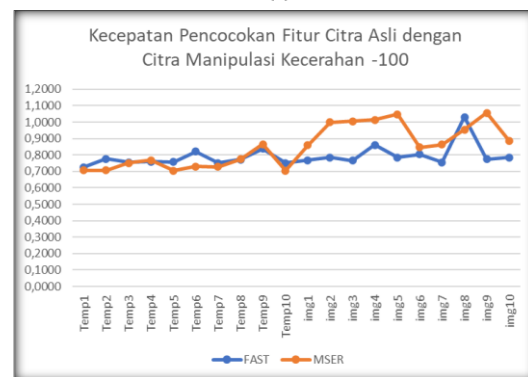
(a)



(b)



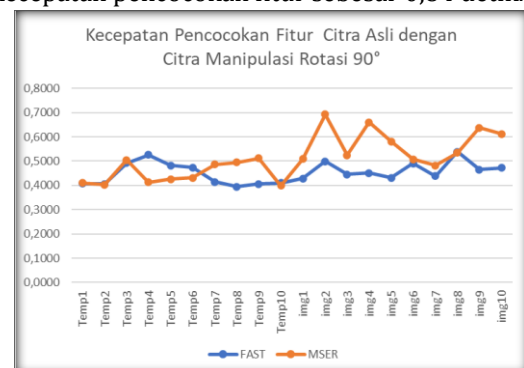
(c)



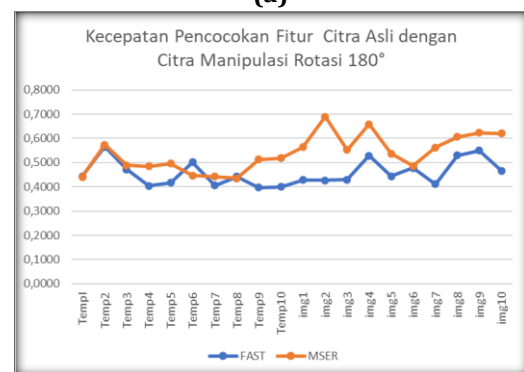
(d)

Gambar 10. Diagram Garis Kecepatan Pencocokan Fitur Citra Asli dengan Citra Manipulasi Kecerahan: (a) +50, (b) +100, (c) -50, (d) -100

Dilihat dari diagram garis diatas, kecepatan pencocokan fitur algoritma FAST terhadap citra asli dengan citra manipulasi kecerahan lebih baik daripada algoritma MSER. Algoritma FAST memiliki *range* antara 0,4–1 detik dan rata-rata kecepatan pencocokan fitur sebesar 0,78 detik. Sedangkan algoritma MSER memiliki *range* antara 0,6–1,1 detik dan rata-rata kecepatan pencocokan fitur sebesar 0,84 detik.



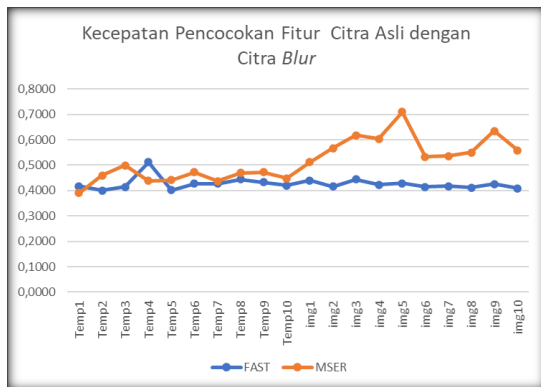
(a)



(b)

Gambar 11. Diagram Garis Kecepatan Pencocokan Fitur Citra Asli dengan Citra Manipulasi Rotasi: (a) 90°, (b) 180°

Berdasarkan diagram garis diatas, kecepatan pencocokan fitur algoritma FAST terhadap citra asli dengan citra manipulasi rotasi lebih baik daripada algoritma MSER. Algoritma FAST memiliki *range* antara 0,4–0,56 detik dan rata-rata kecepatan pencocokan fitur sebesar 0,45 detik. Sedangkan algoritma MSER memiliki *range* antara 0,4–0,7 dan rata-rata kecepatan pencocokan fitur sebesar 0,52 detik.



Gambar 12. Diagram Garis Kecepatan Pencocokan Fitur Citra Asli dengan Citra Blur

Dilihat dari diagram garis diatas, kecepatan pencocokan fitur algoritma FAST terhadap citra asli dengan citra blur lebih baik daripada algoritma MSER. Algoritma FAST memiliki range antara 0,4-0,51 detik dan rata-rata kecepatan pencocokan fitur sebesar 0,42 detik. Sedangkan algoritma MSER memiliki range antara 0,4-0,7 dan rata-rata kecepatan pencocokan fitur sebesar 0,51 detik.

3.1. Rata-Rata Hasil Pengujian

Pengujian dilakukan dengan menguji citra asli terlebih dahulu untuk mendapatkan fitur awal dan akurasi kecocokan dari algoritma FAST dan algoritma MSER terhadap citra asli, serta kecepatan deteksi dan ekstraksi fitur, dan kecepatan pencocokan fitur. Kemudian dilanjutkan dengan mencocokkan fitur citra asli dengan citra hasil manipulasi untuk mendapatkan kemampuan masing-masing algoritma dalam ketahanannya terhadap gangguan pada citra. Berikut hasil pengujian deteksi fitur dan pencocokan fitur terhadap citra asli.

Tabel 6. Hasil Pengujian Deteksi dan Ekstraksi Fitur, dan Pencocokan Fitur Citra Asli Algoritma FAST

Set	Jumlah Fitur	Waktu Ekstraksi Fitur (s)	Fitur Cocok (%)	Waktu Pendeteksian Fitur Cocok (s)
2D				
Template	192	0,097	99,74%	0,4712
Marker				
Image	827	0,0847	97,64%	0,4370
Marker				
Rata-rata	509,5	0,0909	98,69%	0,4541

Tabel 7. Hasil Pengujian Deteksi dan Ekstraksi Fitur, dan Pencocokan Fitur Citra Asli Algoritma MSER

Set	Jumlah Fitur	Waktu Ekstraksi Fitur (s)	Fitur Cocok (%)	Waktu Pendeteksian Fitur Cocok (s)
2D				
Template	192	0,097	99,74%	0,4712
Marker				
Image	827	0,0847	97,64%	0,4370
Marker				
Rata-rata	509,5	0,0909	98,69%	0,4541

Setelah mendapatkan rata-rata persentase akurasi pencocokan fitur citra asli, selanjutnya dilakukan pencocokan fitur antara citra asli dengan citra perubahan kecerahan, citra perubahan rotasi, dan citra dengan *blur*. Berikut rata-rata hasil pencocokan fitur citra asli dengan citra hasil manipulasi.

Tabel 8. Rata-rata Hasil Pengujian Pencocokan Fitur Citra Asli dengan Citra Perubahan Kecerahan

Set	FAST		MSER	
	Fitur Cocok (%)	Waktu (s)	Fitur Cocok (%)	Waktu (s)
2D				
Template	86,93%	0,7568	81,40%	0,7449
Marker				
Image	75,59%	0,8137	72,15%	0,9485
Marker				
Rata-rata	81,26%	0,7852	76,77%	0,8467

Tabel 9. Rata-rata Hasil Pengujian Pencocokan Fitur Citra Asli dengan Citra Perubahan Rotasi

Set	FAST		MSER	
	Fitur Cocok (%)	Waktu (s)	Fitur Cocok (%)	Waktu (s)
2D				
Template	75,39%	0,4429	95,64%	0,4658
Marker				
Image	84,91%	0,4676	87,61%	0,5819
Marker				
Rata-rata	80,15%	0,4552	91,62%	0,5238

Tabel 10. Rata-rata Hasil Pengujian Pencocokan Fitur Citra Asli dengan Citra *Blur*

Set	FAST		MSER	
	Fitur Cocok (%)	Waktu (s)	Fitur Cocok (%)	Waktu (s)
2D				
Template Marker	11,37%	0,4296	56,08%	0,4526
Image Marker	13,96%	0,4230	36,13%	0,5823
Rata-rata	12,66%	0,4263	46,11%	0,5174

4. KESIMPULAN

Berdasarkan penelitian dan pengujian yang telah dilakukan, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Untuk menganalisis kinerja deteksi algoritma FAST dan algoritma MSER dapat dilakukan pengujian eksperimental kedua algoritma terhadap banyaknya fitur citra yang dapat dideteksi dan diekstraksi, kecepatan deteksi dan ekstraksi fitur citra, persentase pencocokan fitur terhadap citra asli dan citra manipulasi, dan kecepatan pencocokan fitur citra menggunakan MATLAB.
2. Berdasarkan hasil analisis kinerja deteksi algoritma FAST dan algoritma MSER terhadap perubahan cahaya, rotasi, dan *blur*, kedua algoritma bekerja dengan baik terhadap perubahan cahaya dan rotasi. Tetapi kedua algoritma tidak dapat bekerja dengan baik terhadap citra *blur*. Namun algoritma MSER memiliki rata-rata persentase pencocokan fitur yang lebih baik daripada algoritma FAST sebesar 71,50%; dengan 76,77% untuk perubahan pencahayaan; 91,62% untuk perubahan rotasi; dan 46,11% untuk *blur*. Sedangkan algoritma FAST memiliki rata-rata persentase pencocokan fitur sebesar 58,02%; dengan 81,26% untuk perubahan pencahayaan; 80,15% untuk perubahan rotasi; dan 12,66% untuk *blur*. Sedangkan untuk kecepatan, algoritma FAST lebih baik daripada algoritma MSER dengan rata-rata kecepatan pencocokan fitur dengan citra manipulasi 0,55 detik; dan 0,62 detik untuk algoritma MSER.
3. Berdasarkan hasil pengujian, untuk mendapatkan pemrosesan yang cepat terhadap deteksi *marker* atau penanda AR, algoritma FAST adalah pilihan yang tepat untuk digunakan pada aplikasi AR. Namun jika

menginginkan pemrosesan deteksi *marker* yang lebih stabil terhadap gangguan-gangguan atau perubahan yang terjadi dapat pada kamera atau *marker* seperti perubahan cahaya, rotasi kamera atau *marker*, dan fokus kamera, maka algoritma MSER menjadi pilihan yang tepat untuk aplikasi AR.

DAFTAR PUSTAKA

- Bhushan Verma, S., & Chandran, S. (2016). Comparative Study of FAST, MSER, and Harris for Palmprint Verification System. *International Journal of Scientific & Engineering Research*, 7(12), 855–858. Diambil dari <http://www.ijser.org>
- Matas, J., Chum, O., Urban, M., & Pajdla, T. (2004). *Robust wide-baseline stereo from maximally stable extremal regions*. 22, 761–767. <https://doi.org/10.1016/j.imavis.2004.02.006>
- Purba, A. M., Harjoko, A., & Wibowo, M. E. (2019). Text Detection In Indonesian Identity Card Based On Maximally Stable Extremal Regions. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 13(2), 177–188. <https://doi.org/10.22146/ijccs.41259>
- Rosten, E., & Drummond, T. (2006). *Machine learning for high-speed corner detection*. 1–14.
- Thanabornvornwiwat, N., & Patanukhom, K. (2018). Marker Registration Technique for Handwritten Text Marker in Augmented Reality Applications. *Journal of Physics: Conference Series*, 1004(1), 1–7. <https://doi.org/10.1088/1742-6596/1004/1/012005>
- Wahyudi, N., Harianto, R. A., & Setyati, E. (2019). Augmented Reality Marker Based Tracking Visualisasi Drawing 2D ke dalam Bentuk 3D dengan Metode FAST Corner Detection. *JOURNAL OF INTELLIGENT SYSTEMS AND COMPUTATION*, 1(1), 9–18.
- Zafar, Z., Berns, K., & Rodić, A. (2017). Recognizing hand gestures using local features: A comparison study. *Advances in Intelligent Systems and Computing*, 540, 394–401. https://doi.org/10.1007/978-3-319-49058-8_43